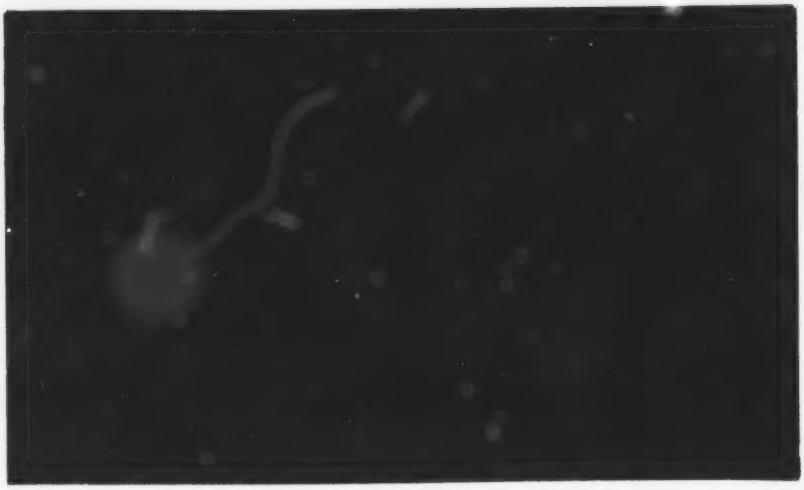
CENTRE
DE RECHERCHE
SUR LES TRANSPORTS
(CRT)
CENTRE
FOR RESEARCH
ON TRANSPORTATION

UNIVERSITÉ DE MONTRÉAL
ÉCOLE DES HAUTES ÉTUDES COMMERCIALES
ÉCOLE POLYTECHNIQUE



C.P. 6128, SUCCURSALE A
MONTRÉAL (QUÉBEC)
CANADA
H3C 3J7

TÉLÉPHONE : (514) 343-7575
TÉLEX : 05-24146 BIBPOLYTEC
TÉLÉCOPIEUR (FAX) : (514) 343-7121

CRT-652

A BRANCH-AND-BOUND APPROACH FOR THE MULTICOMMODITY LOCATION PROBLEM WITH BALANCING REQUIREMENTS

by

Teodor Gabriel Crainic[1]
Louis Delorme[2]
Pierre Dejax[3]

[1] Département des Sciences administratives, Université du Québec à Montréal, DSA-UQAM, C.P. 6192, Succursale "A", Montréal, Québec, Canada H3C 4R2 and Centre de recherche sur les transports, Université de Montréal, C.P. 6128, Succursale "A", Montréal, Québec, Canada H3C 3J7.

[2] I.R.E.Q., Hydro-Québec, Varennes, Québec, Canada.

[3] Laboratoire Économique, Industriel et Social, École Centrale Paris, Chatenay Malabry, France.

This research has been done through a project of cooperation between France and Québec and has been supported by a grant from the F.C.A.R. of Québec and the Natural Sciences and Engineering Research Council of Canada.

Centre de recherche sur les transports - Publication #652

November 1989

90-01617/

Dépôt légal - 4e trimestre 1989

ABSTRACT

We present a branch-and-bound resolution scheme for the multicommodity location–allocation problem with balancing requirements. Although the formulation displays a number of characteristics common to classical location models, it also unveils, due to the presence of the balancing requirements, a network flow structure that is most favorable to efficient algorithmic developments. In particular, tight bounds may be efficiently computed by using a reformulation of the weak relaxation of the problem. We also develop and analyse various branching criteria aimed toward tailoring the most efficient branch-and-bound procedure. It is noteworthy that the class of branching rules that formed the most efficient procedure for the present problem are quite different from that used to solve classical location problems. The experimentation has been conducted both on several randomly generated problems and on a large scale application.

KEY WORDS: MULTICOMMODITY LOCATION–ALLOCATION MODELS,
EMPTY FLOW BALANCING, FLEET MANAGEMENT,
CONTAINER TRANSPORTATION,
BRANCH-AND-BOUND ALGORITHMS.

RESUME

Nous présentons un algorithme d'énumération de type ''branch-and-bound'' pour le problème de localisation–allocation avec échanges inter–dépôts (ou exigences d'équilibrage). Ce modèle présente des similitudes avec les modèles classiques de localisation. Toutefois, dû aux contraintes d'équilibrage, la formulation exhibe aussi une structure de réseau, très favorable aux développements algorithmiques. Ceci permet, en particulier, le calcul efficace de bonnes bornes aux noeux de l'arbre d'énumération, par la résolution de la relaxation faible du problème. Plusieurs critères de branchement sont aussi proposés et analysés, et on remarque que ceux qui contribuent à former l'algorithme le plus efficace pour le problème étudié sont différents des critères utilisés pour résoudre les problèmes classiques de localisation. L'expérimentation a été conduite sur un ensemble de problèmes générés aléatoirement et sur une application de grande taille.

MOTS CLES: MODELES DE LOCALISATION–ALLOCATION MULTIPRODUITS,
GESTION DE FLOTTE, MOUVEMENTS A VIDE,
TRANSPORT DES CONTENEURS,
METHODES DE ''BRANCH-AND-BOUND''.

INTRODUCTION

One of the major logistics issues faced by transportation firms concerns the strategic/tactical aspect of the fleet management problem: the location of the depots for the rolling stock and the forecast and management of the empty movements. A particularly challenging version of this problem is faced by maritime container transportation companies relative to the planning of their land operations. Here follows a very schematic statement of this problem (for a full description and discussion, see Dejax, Crainic and Delorme, 1988):

> The container transportation firm delivers to its clients loaded containers of various types, by using various transportation modes. Following delivery and unloading by the client, empty containers are returned to a depot. The same, or other, clients are also requesting empty containers, of specific types, for subsequent shipping of their own products. Empty containers are delivered from the depots, are loaded, and are subsequently moved, by land and maritime transportation means, to their final destinations. A major complication of the problem is due to the regional imbalances in the empty container availabilities and needs throughout the network: some areas lack container of certain types, while others have surpluses of them. This requires the shipment of empty containers between depots to counter this tendency and thus ''balance the network''.

It is assumed that interdepot transportation is more efficient, per unit, that movements between depots and clients, due to the larger volumes of traffic that may be consolidated (e.g. unit trains) for transportation over relatively long distances. This also justifies the very existence of interdepot movements, and thus differentiates this problem from a classical location/allocation application.

The general problem is therefore to locate the empty container depots in order to collect from clients the supply of empty containers and to satisfy the client demands for empty containers, while minimizing the total operation cost: the cost of opening and operating the depots, plus the transportation costs of the empty movements between depots and between clients and depots. It may be interesting to note that, in practice, this strategic/tactical logistics problem may have to be repeatedly solved since, often, container transportation companies do not build their own depots, but rather use facilities available from other modes (e.g. ports

and rail yards). It is then a question of deciding which facilities to use, according to the demand and the cost structure particular to the specific planning period.

The usual hypothesis concerning this problem assume that depot capacity does not restrict depot utilization, and that the transportation and depot operation costs are linear. The first assumption may be easily verified in actual practice, while the second reflects the available data and costing definitions.

We have already proposed (Crainic, Dejax and Delorme, 1989) a modeling framework for this class of problems, that we call the *simple multicommodity location/allocation problem with balancing requirements*, and that we note (SLBP). Here, *simple* does not imply a "single echelon" formulation since, in fact, the problem may be described as being "two echelon". It rather refers to the fact that in the current formulation costs are linear, supplies and demands are deterministic and both depots and transportation links are uncapacitated. The presence of the balancing requirements represents one of the major differences between this formulation and classical location models, since it unveils a network flow structure that is most favorable to efficient algorithmic developments. Our approach is based on this fact.

The objective of this paper is to present a branch-and-bound algorithm for the SLBP, based on and taking advantage of this particular structure. The paper is organized as follows. The general formulation is recalled in the first section, accompanied by a few comments on comparing the SLBP to the classical simple plant location problem, especially concerning the branch-and-bound approach. Section two presents how to compute efficiently lower and upper bounds on the objective value of the subproblem associated to a node of the branch-and-bound tree, while various branching criteria are discussed in Section three. The experimentation has been conducted both on a wide gamut of randomly generated problems and on a large scale application, and has been aimed toward tailoring the most efficient branch-and-bound procedure out of the preceding elements. Section four presents the results of this experimentation.

I. THE MODEL

The model is formulated (Crainic, Dejax and Delorme, 1989) as:

$$\textit{Minimize}\quad Z = \sum_{j\in D} f_j\, y_j + \sum_{p\in P}\left\{\sum_{i\in C}\sum_{j\in D}\left(c_{ijp}\, x_{ijp} + c_{jip}\, x_{jip}\right) + \sum_{j\in D}\sum_{k\in D} s_{jkp}\, w_{jkp}\right\}$$

$$\textit{Subject to}\quad \sum_{j\in D} x_{ijp} = O_{ip} \qquad \forall\, i\in C, p\in P$$

$$\sum_{j\in D} x_{jip} = D_{ip} \qquad \forall\, i\in C, p\in P \qquad (1)$$

$$x_{ijp} \le M\, y_j \qquad \forall\, i\in C, j\in D, p\in P$$

$$x_{jip} \le M\, y_j \qquad \forall\, i\in C, j\in D, p\in P \qquad (2)$$

$$w_{jkp} \le M\, y_j \qquad \forall\, j\in D, k\in D, p\in P$$

$$w_{kjp} \le M\, y_j \qquad \forall\, j\in D, k\in D, p\in P \qquad (3)$$

$$\sum_{i\in C} x_{ijp} + \sum_{k\in D} w_{kjp} - \sum_{i\in C} x_{jip} - \sum_{k\in D} w_{jkp} = 0 \qquad \forall\, j\in D, p\in P \qquad (4)$$

$$x_{ijp} \ge 0 \qquad \forall\, i\in C, j\in D, p\in P$$

$$x_{jip} \ge 0 \qquad \forall\, i\in C, j\in D, p\in P$$

$$w_{jkp} \ge 0 \qquad \forall\, j\in D, k\in D, p\in P$$

$$y_j \in \{0,1\} \qquad \forall\, j\in D$$

where

C : Set of clients;

D : Set of candidate depots;

P : Set of products;

f_j : "Fixed" cost of depot j, $j\in D$;

c_{ijp}, c_{jip} : Unit transportation costs for product $p, p\in P$, between client $i, i\in C$, and depot $j, j\in D$;

s_{jkp}, s_{kjp} : Unit transportation costs for product $p, p\in P$, between depots j and $k, j\in D, k\in D$;

$y = (y_j)$: Opening vector for depots, $j \in D$,

$$y_j = \begin{cases} 1 & \text{if depot } j \text{ is open} \\ 0 & \text{otherwise} \end{cases}$$

x_{ijp}, x_{jip} : Flows of product $p, p \in P$, between client $i, i \in C$, and depot $j, j \in D$;

w_{jkp}, w_{kjp} : Flows of product $p, p \in P$, between depots j and k, $j \in D, k \in D$;

O_{ip} : Supply at node (client) $i, i \in C$, of product $p, p \in P$;

D_{ip} : Demand at node (client) $i, i \in C$, of product $p, p \in P$;

Constraints (1) ensure that demand and supply requirements are met. Constraints (2) forbid exchanges between clients and closed depots (M is a suitably large positive constant), while constraints (3) have a similar role for interdepot movements. Constraints (4) correspond to the interdepot balancing requirements.

As mentioned in the introduction, we assume that for the range of distances characterizing them, interdepot movements are more efficient, per transported unit, than movements between clients and depots. This forms the rationale of the problem and contributes toward the interest and originality of the model, by insuring that interdepot movements do take place. Yet, it is also reasonable to assume, since it may quite easily be verified in actual practice, that interdepot transportation costs verify the triangle inequality

$$s_{jkp} \leq s_{jrp} + s_{rkp} \quad \forall\, j \in D, k \in D, r \in D$$

Crainic, Dejax and Delorme (1989) show then that constraints (3) are redundant and that the model is more concisely written as:

$$(L)\ \textit{Minimize}\quad Z = \sum_{j\in D} f_j\, y_j + \sum_{p\in P}\left\{\sum_{i\in C}\sum_{j\in D}\left(c_{ijp}\, x_{ijp} + c_{jip}\, x_{jip}\right) + \sum_{j\in D}\sum_{k\in D} s_{jkp}\, w_{jkp}\right\}$$

Subject to

$$\sum_{j\in D} x_{ijp} = O_{ip} \qquad \forall\, i\in C, p\in P$$

$$\sum_{j\in D} x_{jip} = D_{ip} \qquad \forall\, i\in C, p\in P \qquad \text{(L1)}$$

$$x_{ijp} \le O_{ip}\, y_j \qquad \forall\, i\in C, j\in D, p\in P$$

$$x_{jip} \le D_{ip}\, y_j \qquad \forall\, i\in C, j\in D, p\in P \qquad \text{(L2)}$$

$$\sum_{i\in C} x_{ijp} + \sum_{k\in D} w_{kjp} - \sum_{i\in C} x_{jip} - \sum_{k\in D} w_{jkp} = 0 \qquad \forall\, j\in D, p\in P \qquad \text{(L3)}$$

$$x_{ijp} \ge 0 \qquad \forall\, i\in C, j\in D, p\in P$$

$$x_{jip} \ge 0 \qquad \forall\, i\in C, j\in D, p\in P \qquad \text{(L4)}$$

$$w_{jkp} \ge 0 \qquad \forall\, j\in D, k\in D, p\in P \qquad \text{(L5)}$$

$$y_j \in \{0,1\} \qquad \forall\, j\in D \qquad \text{(L6)}$$

When this model is compared to classical location formulations (Aikens, 1985), a number of similitudes and differences may be quite readily identified. Since only one level of facilities is to be located, SLBP is close to the simple plant location problem - SPLP - (Krarup and Pruzan, 1983). Yet, the problem may also be described as a two echelon problem, from plants, through depots, to clients, with the requirements that depot workloads be equilibrated and all supply be shipped out of plants. However, in this case, plants do not have to be located, since clients are present at the supply echelon. Moreover, this is a true multi-commodity problem, where sole sourcing is not required for any commodity.

Most significantly, however, the SLBP does not possess the *single assignment* property, contrary to the single plant location problem. This characteristic is mostly due to the existence of interdepot flows, and influences significantly the definition of the branch-and-bound procedure. Yet, there are sufficient similarities between the two problems, to allow us to consider the results obtained while developing enumeration type algorithm for the SPLP (see, Krarup and Pruzan 1983, for a review of these issues), especially the classical results of Khumawala (1972) concerning the development of efficient branching rules for the branch-and-bound.

The branch-and-bound algorithm is a very well known solution method for integer and mixed integer problems (Nemhauser and Wolsey, 1988; Ibaraki, 1987). Not necessarily the most efficient method, it offers, however, an algorithmic framework which guarantees that an optimal solution will eventually be found. The quality of the branch-and-bound algorithm is determined by the choices made concerning the computation of bounds on the optimal solution of the subproblem generated at each of the nodes of the branch-and-bound tree, and the rules by which the next subproblem is generated. The presentation of these rules in the present context forms the subject of the two following sections.

II. BOUNDING PROCEDURES

Each subproblem that may be generated during the branch-and-bound enumeration, may be characterized by the *state* that has been decided for each one of the depots: *open, closed* or *undecided (or free)* yet. For the current subproblem, let

$K_0 = \{ j \mid y_j = 0, j \in D \}$: the set of all closed depots

$K_1 = \{ j \mid y_j = 1, j \in D \}$: the set of all open depots

and $K_{01} = \{ j \mid y_j \notin K_0 \cup K_1, j \in D \}$: the set of all free depots.

A *lower bound* on the value of the optimal solution of the subproblem that corresponds to each node of the branch-and-bound tree, may then be computed by solving the *weak relaxation* of the problem:

$$(WR)\ \textit{Minimize}\ Z = \sum_{j \in K_1} f_j + \sum_{j \in K_{01}} f_j\, y_j + \sum_{p \in P} \left\{ \sum_{i \in C} \sum_{j \in K_1 \cup K_{01}} (c_{ijp}\, x_{ijp} + c_{jip}\, x_{jip}) + \sum_{j \in K_1 \cup K_{01}} \sum_{k \in K_1 \cup K_{01}} s_{jkp}\, w_{jkp} \right\}$$

$$\textit{Subject to} \sum_{j \in K_1 \cup K_{01}} x_{ijp} = O_{ip} \quad \forall\, i \in C, p \in P$$

$$\sum_{j \in K_1 \cup K_{01}} x_{jip} = D_{ip} \quad \forall\, i \in C, p \in P \qquad \text{(WR1)}$$

$$\sum_{p \in P} \sum_{i \in C} (x_{ijp} + x_{jip}) \le \sum_{p \in P} \sum_{i \in C} (O_{ip} + D_{ip}) \quad \forall\, j \in K_1 \qquad \text{(WR2)}$$

$$\sum_{p \in P} \sum_{i \in C} (x_{ijp} + x_{jip}) \le \left[\sum_{p \in P} \sum_{i \in C} (O_{ip} + D_{ip}) \right] y_j \quad \forall\, j \in K_{01} \qquad \text{(WR3)}$$

$$\sum_{i \in C} x_{ijp} + \sum_{k \in K_1 \cup K_{01}} w_{kjp} - \sum_{i \in C} x_{jip} - \sum_{k \in K_1 \cup K_{01}} w_{jkp} = 0 \quad \forall\, j \in K_1 \cup K_{01}, p \in P \qquad \text{(WR4)}$$

$$x_{ijp} = 0 \quad \forall\, i \in C, j \in K_0, p \in P$$

$$x_{jip} = 0 \quad \forall\, i \in C, j \in K_0, p \in P \qquad \text{(WR5)}$$

$$x_{ijp} \ge 0 \quad \forall\, i \in C, j \in K_1 \cup K_{01}, p \in P$$

$$x_{jip} \ge 0 \quad \forall\, i \in C, j \in K_1 \cup K_{01}, p \in P \qquad \text{(WR6)}$$

$$w_{jkp} = 0 \quad \forall\, j \in K_0, k \in D, p \in P \text{ or } j \in D, k \in K_0, p \in P \qquad \text{(WR7)}$$

$$w_{jkp} \geq 0 \qquad \forall\, j \in K_1 \cup K_{01},\ k \in K_1 \cup K_{01},\ p \in P \qquad \text{(WR8)}$$

$$y_j \geq 0 \qquad \forall\, j \in K_1 \cup K_{01} \qquad \text{(WR9)}$$

It is usually admitted that lower bounds computed by using the weak relaxation of a linear integer problem are generally less "good", by being less "tight", that those obtained by using the strong relaxation (where constraints (L2) are not added up in the relaxation) of the same problem. It is also true, however, that the weak relaxation is often more easy to solve, as for the SPLP for example, than the corresponding strong relaxation problem. Then, the former bounding procedure is preferred for inclusion in branch-and-bound resolution methods. This is particularly the case for the present problem, where the weak relaxation formulation presents properties particularly interesting from an algorithmic point of view. First, it is quite trivial to prove that

Proposition 1. Any optimal solution to problem (WR) satisfies constraints (WR3) with strict equality.

Consequently, for a free depot j, $j \in K_{01}$, the value of the associated y_j variable

$$y_j = \frac{\sum_{p \in P} \sum_{i \in C} (x_{ijp} + x_{jip})}{\sum_{p \in P} \sum_{i \in C} (O_{ip} + D_{ip})}$$

represents the proportion of the total traffic demand (total client supply and demand) which is satisfied through depot j. By using this observation, the weak relaxation formulation becomes:

$$(WRF)\ \ Minimize\ \ Z = \sum_{j \in K_1} f_j + \sum_{p \in P} \left\{ \sum_{i \in C} \sum_{j \in K_1} (c_{ijp}\, x_{ijp} + c_{jip}\, x_{jip}) \right.$$

$$+ \sum_{i \in C} \sum_{j \in K_{01}} \left[\left(c_{ijp} + \frac{f_j}{\Delta} \right) x_{ijp} + \left(c_{jip} + \frac{f_j}{\Delta} \right) x_{jip} \right]$$

$$\left. + \sum_{j \in K_1 \cup K_{01}} \sum_{k \in K_1 \cup K_{01}} s_{jkp}\, w_{jkp} \right\}$$

$$\text{Subject to} \sum_{j \in K_1 \cup K_{01}} x_{ijp} = O_{ip} \qquad \forall\, i \in C,\ p \in P$$

$$\sum_{j \in K_1 \cup K_{01}} x_{jip} = D_{ip} \qquad \forall\, i \in C,\ p \in P \tag{WF1}$$

$$\sum_{i \in C} x_{ijp} + \sum_{k \in K_1 \cup K_{01}} w_{kjp} - \sum_{i \in C} x_{jip} - \sum_{k \in K_1 \cup K_{01}} w_{jkp} = 0 \quad \forall\, j \in K_1 \cup K_{01},\ p \in P \tag{WF2}$$

$$x_{ijp} \geq 0 \qquad \forall\, i \in C,\ j \in K_1 \cup K_{01},\ p \in P$$

$$x_{jip} \geq 0 \qquad \forall\, i \in C,\ j \in K_1 \cup K_{01},\ p \in P \tag{WF3}$$

$$w_{jkp} = 0 \qquad \forall\, j \in K_0,\ k \in D,\ p \in P \text{ or } j \in D,\ k \in K_0,\ p \in P \tag{WF4}$$

$$w_{jkp} \geq 0 \qquad \forall\, j \in K_1 \cup K_{01},\ k \in K_1 \cup K_{01},\ p \in P \tag{WF5}$$

where $\Delta = \sum_{p \in P} \sum_{i \in C} (O_{ip} + D_{ip})$. One may also easily prove the following:

Proposition 2. At any nonterminal node of the branch-and-bound tree, the optimal solution to problem (WRF) ensures that $\sum_{j \in K_{01}} y_j \leq 1$, i.e. that $0 \leq y_j \leq 1 \ \ \forall\, j \in K_{01}$.

The next property (see Crainic, Dejax and Delorme, 1989 for the proof) justifies the choice of the weak relaxation formulation to compute bounds at the nodes of the branch-and-bound tree:

Proposition 3. (WRF) is an uncapacitated minimum cost, multi-commodity flow problem.

Moreover, since each constraint of (WRF) is product specific, this formulation may be partitioned into π mutually independent minimum cost flow problems, where π is defined as the cardinality of P, which may then be solved sequentially. Or, as several very efficient solution algorithms exist for the minimum cost flow problem, we use the (WRF) formulation to compute the lower bounds for the branch-and-bound algorithm.

An *upper bound* on the optimal solution of (L) may be computed by fixing to one the value of all depot variables in the free depot set. This bound may be rather loose, especially in the early stages of the algorithm. It may be tightened when an integer solution is reached at a node of the branch-and-bound tree, since that solution is feasible for the original problem, as formally stated in the following proposition:

Proposition 4. If the optimal solution of the (WRF) problem, corresponding to a node of the branch-and-bound tree, results in integer values for the variables associated to the free depots, then:

(i) the solution is feasible for the original problem (L);

(ii) the solution is optimal for problem (L) where all y_j variables corresponding to open and closed depots are fixed at their values.

To prove the first part, note that closed depots imply zero flows on the arcs originating and terminating at those depots. Then, constraints (L1) and (L3) are identical to constraints (WF1) and (WF2), respectively. Constraints (L2) are equally verified since, by non-negativity and constraints (WF1), $x_{ijp} \leq O_{ip}$ and $x_{jip} \leq D_{ip}$ for all $i \in C$, $p \in P$ and $j \in K_1 \cup K_{01}$, while for closed depots, the corresponding traffic is zero. Finally, non-negativity constraints are equivalent in the two formulations, while the integrality conditions are verified by hypothesis.

To verify the second part of the proposition, we only have to substitute the solution to problem (WRF) in the objective function of problem (L) and show that we obtain the objective of (WRF). First, let (y', x', w') be the optimal solution to (WRF). By Proposition 1 we know that

$$y_j' = \frac{\sum_{p \in P} \sum_{i \in C} (x_{ijp}' + x_{jip}')}{\sum_{p \in P} \sum_{i \in C} (O_{ip} + D_{ip})}$$

for all $j \in K_{01}$. But, y' is integer; therefore:

- If $y_j' = 0$ then the corresponding $x_{ijp}' = x_{jip}' = 0$ for all $i \in C$ and $p \in P$.
- If, on the other hand, there is a $j^* \in K_{01}$ such as $y_{j^*}' = 1$, then all flow passes through that depot and $x_{ij^*p}' = O_{ip}$, $x_{j^*ip}' = D_{ip}$, $\forall\, i \in C$, $p \in P$, and $y_j' = 0$, $\forall\, j \in K_{01}$, $j \neq j^*$.

The objective function of problem (WRF) may then be rewritten as:

$$Z = \sum_{j \in K_1} f_j + \sum_{p \in P} \left\{ \sum_{j \in K_1 \cup \{j^*\}} \sum_{k \in K_1 \cup \{j^*\}} s_{jkp}\, w_{jkp}' \right.$$

$$\left. + \sum_{i \in C} \sum_{j \in K_1} (c_{ijp}\, x_{ijp}' + c_{jip}\, x_{jip}') + \left[(c_{ij^*p} + \frac{f_{j^*}}{\Delta}) O_{ip} + (c_{j^*ip} + \frac{f_{j^*}}{\Delta}) D_{ip} \right] \right\}$$

$$= \sum_{j \in K_1} f_j + \sum_{p \in P} \left\{ \sum_{j \in K_1 \cup \{j^*\}} \sum_{k \in K_1 \cup \{j^*\}} s_{jkp} \, w'_{jkp} \right.$$

$$\left. + \sum_{i \in C} \sum_{j \in K_1} (c_{ijp} \, x'_{ijp} + c_{jip} \, x'_{jip}) + \left[\frac{f_{j^*}}{\Delta} (O_{ip} + D_{ip}) + (c_{ij^*p} \, O_{ip} + c_{j^*ip} \, D_{ip}) \right] \right\}$$

$$= \sum_{j \in K_1 \cup \{j^*\}} f_j + \sum_{p \in P} \left\{ \sum_{j \in K_1 \cup \{j^*\}} \sum_{k \in K_1 \cup \{j^*\}} s_{jkp} \, w'_{jkp} \right.$$

$$\left. + \sum_{i \in C} \sum_{j \in K_1} (c_{ijp} \, x'_{ijp} + c_{jip} \, x'_{jip}) + (c_{ij^*p} \, O_{ip} + c_{j^*ip} \, D_{ip}) \right\}$$

which is exactly the objective function of problem (L) evaluated at the current solution.

□

III. BRANCHING RULES

Branching rules determine how to select the next active node and which one of the two possible branches to explore first. The efficiency of the branch-and-bound algorithm depends largely on the sharpness of these rules and their capability to assist the pruning of the enumeration tree by improving the bounds and fixing variables to reach feasible integer solutions more rapidly.

Some of the rules presented in this section are derived from the pioneering work of Khumawala (1972), in particular the *Omega, Delta* and *Y* rules. Yet, we do not expect these rules to display performances similar to those observed on the SPLP, since the SLBP does not possess the single assignment property.

Y Rules

Similar to Khumawala's, these rules select the variable y_j', $j \in K_{01}$, which is closest to either 0 or 1, and treats first the branch where the variable is fixed to this value. The intuitive argument supporting this rule is that y_j' which is close to 0 (1) has a "better" chance of being closed (open) in the best solution to be found in the current node's subtree.

Modified Y Rules

In addition to the previous considerations on the values of the variables y_j', $j \in K_{01}$, it may be assumed that a depot's opening and operating costs may also play a significant role in deciding to open or to close it. Specifically, it is inferred that a depot with a high opening cost is more likely to be closed in the best solution of the subtree. The coefficient which captures the effect of these two observations is defined as

$$Y_j^{mod} = f_j \ (1 - y_j') \qquad \forall \ j \in K_{01}$$

and is used to specify the two following rules:

(1) *Largest Modified Y Rule (LMY)*. Select the free depot which has the maximum Y_j^{mod} and explore first the branch where the depot is closed.

(2) *Smallest Modified Y Rule (SMY)*. Select the free depot which has the minimum Y_j^{mod} and explore first the branch where the depot is open.

Delta Rules

A lower bound on the impact of opening a depot, which currently is in the set of free depots, may be quite easily evaluated when solving the SPLP by a branch-and-bound algorithm, since this problem displays the single assignment property. This bound is obtained by computing the minimum cost savings for each client due to the opening of the depot and, then, by comparing the resulting total savings to the depot opening cost. The bound is then used both to simplify the problem (by fixing open depots with positive bounds) and to define two branching rules (Khumawala, 1972).

When balancing requirements are considered, a similar bound can not be obtained without a considerable computation effort involving, in particular, the resolution of several multicommodity minimum cost flow problems. We therefore approximate these measures by

$$\nabla_{ijp} = \min_{k \in K_1 \cup K_{01},\, k \neq j} \{ \max \{ (c_{kip} D_{ip} + c_{ikp} O_{ip}) - (c_{jip} D_{ip} + c_{ijp} O_{ip}), 0 \} \}$$

$$\forall j \in K_{01},\ i \in C,\ p \in P$$

and

$$\Delta_j = \sum_{i \in C} \sum_{p \in P} \nabla_{ijp} - f_j \qquad \forall j \in K_{01}$$

Two rules may then be defined:

(1) *Largest Delta Rule (LΔ)*. Select the free depot which has the maximum positive Δ_j, and explore first the branch where the depot is open. If $\Delta_j < 0$, $\forall j \in K_{01}$, then apply the $S\Delta$ rule.

(2) *Smallest Delta Rule (SΔ)*. Select the free depot which has the minimum negative Δ_j, and explore first the branch where the depot is closed. If $\Delta_j > 0$, $\forall j \in K_{01}$, then apply the $L\Delta$ rule.

The preceding rules do not take into account the fact that, for the SLBP, not all of a client's demand is shipped out from one depot, nor all of a client's supply is sent to a unique depot. Therefore, an alternate criteria may be defined as:

$$\nabla_{ijp}^{mod} = \min_{k \in K_1 \cup K_{01},\, k \neq j} \{ \max \{ (c_{kip}\, x_{kip} + c_{ikp}\, x_{ikp}) - (c_{jip}\, x_{jip} + c_{ijp}\, x_{ijp}), 0 \} \}$$

$$\forall\, j \in K_{01},\, i \in C,\, p \in P$$

and

$$\Delta_j^{mod} = \sum_{i \in C} \sum_{p \in P} \nabla_{ijp}^{mod} - f_j \qquad \forall\, j \in K_{01}$$

and the corresponding rules are:

(1) *Largest Modified Delta Rule (LMΔ)*. Select the free depot which has the maximum positive Δ_j^{mod}, and explore first the branch where the depot is open.
If $\Delta_j^{mod} < 0$, $\forall\, j \in K_{01}$, then apply the *SMΔ* rule.

(2) *Smallest Modified Delta Rule* (SMΔ) Select the free depot which has the minimum negative Δ_j^{mod}, and explore first the branch where the depot is closed.
If $\Delta_j^{mod} > 0$, $\forall\, j \in K_{01}$, then apply the *LMΔ* rule.

Omega Rules

The *Omega* criteria are equally inspired by similar rules developed for the SPLP by Khumawala (1972), and the same general considerations and restrictions mentioned for the *Delta Rules* are equally valid here. For the SPLP, the *Omega Rules* evaluate an upper bound on the impact of opening a depot currently in the free depot set. First, the maximum savings for each client, due to the opening of the depot, are computed. The total client savings are then compared to the depot opening cost, the resulting measure being used to either simplify the problem (by closing depots with bounds of negative value) or to branch. The computations are similar to those of the *Delta Rules*, but are carried exclusively on the depots already fixed open. An approximation of these measures for the SLBP is

$$\omega_{ijp} = \min_{k \in K_1} \{ \max \{ (c_{kip}\, D_{ip} + c_{ikp}\, O_{ip}) - (c_{jip}\, D_{ip} + c_{ijp}\, O_{ip}), 0 \} \}$$

$$\forall\, j \in K_{01},\, i \in C,\, p \in P$$

and

$$\Omega_j = \sum_{i \in C} \sum_{p \in P} \omega_{ijp} - f_j \qquad \forall\, j \in K_{01}$$

and the two branching rules are:

(1) *Largest Omega Rule (LΩ)*. Select the free depot which has the maximum positive Ω_j, and explore first the branch where the depot is open. If $\Omega_j < 0, \quad \forall\, j \in K_{01}$, then apply the $S\Omega$ rule.

(2) *Smallest Omega Rule (SΩ)*. Select the free depot which has the minimum negative Ω_j, and explore first the branch where the depot is closed.
If $\Omega_j > 0, \quad \forall\, j \in K_{01}$, then apply the $L\Omega$ rule.

Similarly to the *Modified Delta Rules* and according to the same arguments, we also define

$$\omega_{ijp}^{mod} = \min_{k \in K_1} \{ \max \{ (c_{kip}\, x_{kip} + c_{ikp}\, x_{ikp}) - (c_{jip}\, x_{jip} + c_{ijp}\, x_{ijp}), 0 \} \}$$

$$\forall\, j \in K_{01},\, i \in C,\, p \in P$$

and

$$\Omega_j^{mod} = \sum_{i \in C} \sum_{p \in P} \omega_{ijp}^{mod} - f_j \qquad \forall\, j \in K_{01}$$

and the corresponding rules:

(1) *Largest Modified Omega Rule (LMΩ)*. Select the free depot which has the maximum positive Ω_j^{mod}, and explore first the branch where the depot is open.
If $\Omega_j^{mod} < 0, \quad \forall\, j \in K_{01}$, then apply the $SM\Omega$ rule.

(2) *Smallest Modified Omega Rule (SMΩ)*. Select the free depot which has the minimum negative Ω_j^{mod}, and explore first the branch where the depot is closed.
If $\Omega_j^{mod} < 0, \quad \forall\, j \in K_{01}$, then apply the $LM\Omega$ rule.

All previous rules ignored the effect of the interdepot transportation costs on the locational decision and concentrated exclusively on the cost of transportation between clients and depots. The following three rules attempt to address this issue by comparing some measure of the transportation cost ''associated'' with a depot to the total traffic between this depot and the clients. The rationale that supports these rules assumes that the traffic between clients and depots forms the ''profitable'' part of the company's operations and that a depot with a high (low) ratio of transportation costs to moved flows is more likely to be closed (open) in the

best solution reached from the current node.

Quotient Rules

These rules compare the balancing costs to the client traffic for each depot currently in the set of free depots, by computing

$$Q_j = \frac{\sum_{k \in D} \sum_{p \in P} (w'_{kjp} \, s_{kjp} + w'_{jkp} \, s_{jkp})}{\sum_{i \in C} \sum_{p \in P} (x'_{ijp} + x'_{jip})} \qquad \forall \, j \in K_{01}$$

Two branching rules may be defined:

(1) *Largest Quotient Rule (LQ)*. Select the free depot which has the maximum Q_j, and explore first the branch where the depot is closed. If $Q_j = 0, \quad \forall \, j \in K_{01}$, select then the free depot which corresponds to $\min_{j \in K_{01}} \sum_{i \in C} \sum_{p \in P} (x'_{ijp} + x'_{jip})$.

(2) *Smallest Quotient Rule (SQ)*. Select the free depot which has the minimum Q_j, and explore first the branch where the depot is open. If $Q_j = 0, \quad \forall \, j \in K_{01}$, select then the free depot which corresponds to $\max_{j \in K_{01}} \sum_{i \in C} \sum_{p \in P} (x'_{ijp} + x'_{jip})$.

Modified Quotient Rules

These rules compare the opening and the balancing costs to the total traffic between the clients and a depot currently in the set of free depots, by computing

$$N_j = \frac{f_j + \sum_{k \in D} \sum_{p \in P} (w'_{kjp} \, s_{kjp} + w'_{jkp} \, s_{jkp})}{\sum_{i \in C} \sum_{p \in P} (x'_{ijp} + x'_{jip})} \qquad \forall \, j \in K_{01}$$

Two criteria are then defined:

(1) *Largest Modified Quotient Rule (LMQ)*. Select the free depot which has the maximum N_j, and explore first the branch where the depot is closed.

(2) *Smallest Modified Quotient Rule (SMQ)*. Select the free depot which has the minimum N_j, and explore first the branch where the depot is open.

Z Rules

These branching criteria compare the total transportation cost "related" to a depot, currently in the set of free depots, to the total flow between this depot and the clients it serves:

$$Z_j = \frac{f_j + \sum_{k \in D} \sum_{p \in P} (w'_{kjp} s_{kjp} + w'_{jkp} s_{jkp}) + \sum_{i \in C} \sum_{p \in P} (c_{ijp} x'_{ijp} + c_{jip} x'_{jip})}{\sum_{i \in C} \sum_{p \in P} (x'_{ijp} + x'_{jip})} \quad \forall j \in K_{01}$$

The rules specify that:

(1) *Largest Z Rule (LZ)*. Select the free depot which has the maximum Z_j, and explore first the branch where the depot is closed.

(2) *Smallest Z Rule (SZ)*. Select the free depot which has the minimum Z_j, and explore first the branch where the depot is open.

Note that the computation of the three preceding measures may be significantly simplified, since:

$$\sum_{p \in P} \sum_{i \in C} (x'_{ijp} + x'_{jip}) = y'_j \sum_{p \in P} \sum_{i \in C} (O_{ip} + D_{ip}) \quad \forall j \in K_{01}$$

Therefore

$$Q_j = \frac{1}{y'_j} \left\{ \sum_{k \in D} \sum_{p \in P} (w'_{kjp} s_{kjp} + w'_{jkp} s_{jkp}) \right\}$$

$$N_j = \frac{1}{y'_j} \left\{ f_j + \sum_{k \in D} \sum_{p \in P} (w'_{kjp} s_{kjp} + w'_{jkp} s_{jkp}) \right\}$$

$$Z_j = \frac{1}{y'_j} \left\{ f_j + \sum_{k \in D} \sum_{p \in P} (w'_{kjp} s_{kjp} + w'_{jkp} s_{jkp}) + \sum_{i \in C} \sum_{p \in P} (c_{ijp} x'_{ijp} + c_{jip} x'_{jip}) \right\}$$

IV. EXPERIMENTATION

The experimentation was conducted on a set of "random" test problems, that we generated, and on data from an actual large scale application. We had to devise our own procedure to generate random test problems, since those already available in the literature lack the interdepot balancing dimension. The Annex presents a detailed description of this procedure. We start this section with the presentation and discussion of the results relative to the performance of the branching criteria. The analysis pertaining to the efficiency of the bounding procedure concludes the section.

The first experiments, aimed at determining the most efficient branching criteria were conducted on several medium-sized problems, comprising some 150 nodes and 26 candidate locations for depots. Tables 1, 2 and 3 illustrate the type of results we obtained. The tables display measures of the relative performance of each one of the branching rules described in the previous section, on nine problems which differ in the number of products considered and the importance of the interdepot traffic required by the differences between supply and demand present in each one of the zones of the network.

Rule	Z1P1	Z1P2	Z1P3	Z1P4	Z1P5	Z2P4	Z2P5	Z4P4	Z4P5
Y	14.70	31.50	14.40	7.80	7.40	7.80	–	–	–
SMY	18.10	36.90	16.40	7.90	7.60	8.40	8.40	11.00	6.30
LMY	13.80	29.60	17.80	–	8.00	9.90	–	11.30	–
SM Δ	20.20	49.70	17.90	9.80	8.10	8.40	9.90	12.40	7.00
LM Δ	–	49.70	17.90	9.80	8.10	8.40	9.90	12.40	7.00
SM Ω	14.20	31.50	13.80	7.60	7.40	7.80	–	–	–
LM Ω	14.20	31.50	13.80	7.60	7.40	7.80	8.40	11.70	6.20
SQ	19.40	44.10	16.90	10.20	7.80	9.80	10.20	12.50	7.00
LQ	15.00	33.50	17.60	–	–	–	–	12.90	–
SMQ	14.80	28.00	14.00	8.50	7.20	7.50	8.20	11.20	6.30
LMQ	13.70	28.60	14.00	–	–	–	9.50	–	–
SZ	14.30	30.70	14.00	8.30	7.40	7.50	8.90	11.20	6.40
LZ	16.80	28.60	15.10	–	–	–	10.00	–	–

Table 1. Performance of Branching Criteria - Relative Gap (%)

The type of the problem is indicated by its label. The figure that follows the Z mark indicates the degree of unbalance present in the system (the higher the figure, the larger the required interdepot flows), while the figure labeled with P corresponds to the number of products. The tables do not show results for the regular *delta* and *omega* rules, since they were consistently outperformed by the corresponding *modified* criteria.

Rule	Z1P1	Z1P2	Z1P3	Z1P4	Z1P5	Z2P4	Z2P5	Z4P4	Z4P5
Y	248	151	409	635	681	741	–	–	–
SMY	950	740	1001	760	712	959	894	761	456
LMY	66	166	230	–	772	997	–	946	–
SM Δ	957	452	954	428	867	613	832	678	749
LM Δ	–	444	954	428	825	613	832	678	749
SM Ω	924	182	631	534	407	268	–	–	–
LM Ω	965	201	429	542	401	131	758	922	137
SQ	715	990	430	952	655	202	860	326	890
LQ	642	380	278	–	–	–	–	641	–
SMQ	830	986	873	347	391	895	820	586	800
LMQ	645	119	428	–	–	–	570	–	–
SZ	935	735	899	916	879	649	844	428	664
LZ	801	177	803	–	–	–	973	–	–

Table 2. Performance of Branching Criteria - "Optimal" Iteration

The measures indicated in the three tables correspond to 1000 iterations of the branch-and-bound algorithm. This number is sufficient for our purposes, since the goal of this experiment is not to find optimal solutions to these problems, but rather to determine which branching criteria are the most efficient in identifying good integer solutions and, thus, in decreasing the upper bound. It is interesting however that, while good solutions were identified quite early during the enumeration, 1000 iterations were not sufficient to completely explore the branch-and-bound tree for any of the problems studied. This is an indication that the lower bound may not be as tight as desired. We further address this topic toward the end of this section.

Rule	Z1P1	Z1P2	Z1P3	Z1P4	Z1P5	Z2P4	Z2P5	Z4P4	Z4P5
Y	474.80 0.08	773.35 0.08	1333.73 0.09	1904.89 0.09	2689.29 0.10	2108.24 0.09	2558.33 0.10	1993.13 0.10	2497.78 0.10
SMY	876.61 0.08	1347.03 0.09	2529.89 0.08	3277.97 0.08	4203.77 0.08	3138.88 0.08	3938.15 0.08	3181.20 0.08	3605.03 0.08
LMY	499.69 0.10	737.56 0.10	1443.34 0.10	2441.98 0.11	2927.76 0.10	2303.30 0.10	2837.20 0.11	2168.25 0.10	3098.87 0.11
SM Δ	801.61 1002.35	1330.63 2767.67	2474.05 3824.73	3230.92 4212.44	4294.66 5592.72	3169.32 5013.85	4061.82 6367.38	3132.80 4950.62	3576.63 4942.47
LM Δ	– –	1332.90 2767.76	2474.05 3815.35	3230.92 4202.60	4310.85 5146.18	3169.32 5001.81	4061.82 6351.96	3122.80 4938.35	3576.63 4930.36
SM Ω	491.32 225.82	781.47 514.65	1494.22 1170.12	2124.98 1988.31	2835.24 1868.75	2249.26 2151.73	2581.13 2478.74	2096.80 2512.28	2832.67 2892.39
LM Ω	499.63 229.02	786.43 432.93	1468.95 1195.21	2132.25 1968.29	2834.37 1416.37	2257.80 1988.43	2742.21 2723.21	2122.62 2243.25	2782.70 2877.92
SQ	876.35 53.22	1372.08 132.59	2536.89 229.84	3304.38 177.55	4175.71 261.13	3153.52 282.47	4049.98 252.87	3150.65 204.39	3648.28 435.52
LQ	488.59 56.36	856.75 118.64	1428.53 176.85	3042.69 724.61	3138.02 644.23	2435.08 422.95	2791.02 556.42	2296.37 348.70	2988.14 596.57
SMQ	855.23 82.75	1294.55 178.18	2430.21 255.92	3178.81 196.27	4137.04 312.46	3131.39 323.00	3940.22 358.09	3148.75 275.20	3632.40 435.57
LMQ	488.76 51.41	797.94 131.81	1447.11 234.92	2309.40 457.44	2925.68 500.44	2204.38 436.32	2628.02 330.10	2205.83 393.36	2901.49 601.11
SZ	855.41 91.33	1326.95 191.52	2480.06 281.23	3198.61 182.86	4197.77 274.03	3082.12 318.92	3936.03 429.99	3157.92 280.38	3636.02 433.68
LZ	511.38 75.68	799.09 137.62	1573.16 231.46	2847.90 686.99	3128.60 641.74	2269.58 403.54	2718.47 444.24	2144.01 413.64	2905.48 657.62

Table 3. Performance of Branching Criteria - Computation Times (CPU sec.)

For each criteria and for each problem, the data in Table 1 gives the relative gap (in percentage) between the best solution found and the initial lower bound. It serves as a measure of the relative precision of each method and of the "quality" of solution it may reach, given a limited amount of computing effort. Table 2 shows the number of the iteration where the best solution has been found. It is a first measure of efficiency, since it indicates how many nodes of the branch-and-bound tree one has to explore before attaining the "optimal" (with respect to the current optimality criteria: all the tree is either explored or pruned, or the maximum number of iterations is reached) solution. In the first two tables, a – indication in the entry for a given rule indicates that no feasible solution has been found in 1000 iterations of

the algorithm. Table 3 displays a second measure of efficiency: how much computing effort is needed to prove that the solution is optimal. Two figures are shown. The first is the total time required to treat the nodes of the tree: generate and store them, compute the bounds, verify the optimality, etc. The second figure corresponds to the time needed to perform the computations required by the branching criteria and to select the next free variable. The tests have been carried out on a 286 personal computer, operating at 16 Mhz; all programs are written in FORTRAN.

The performance of each rule is evaluated according to the quality of the solution found and to the efficiency of the search. Several conclusions then emerge from our experiments, as illustrated in Tables 1, 2 and 3:

- The *delta* rules, modified or not, are totally inefficient, displaying the worst performance on all problems.

- The *omega* rules display better performances, similar to those of the Y rule, as far as the quality of the solution is concerned. This similarity is to be expected, since the Y rule is used in the first stages of the algorithm when no depot is yet fixed open. However, the total times are constantly higher for the *omega* rules than for all other criteria (except the *delta* rules, of course). Thus, since the same quality of solution may be more efficiently obtained by using other selection criteria, the *omega* branching rules are not to be used when solving the SLBP. This confirms that the behaviour of the branch-and-bound algorithm is indeed different for the SLBP and the classical SPLP, since Khumawala (1972) identified the *omega* rule as the most efficient for this problem.

- The branching rules that close the selected free depot (i.e. set $y_j' = 0$; rules LMY, $SM\Delta$, $SM\Omega$, LQ, LMQ, LZ) are generally more efficient, especially in terms of computing times. This is due to the fact that, once a depot is fixed closed, the problem is simplified and the size of the network is diminished, since all arcs between the depot and the clients may be dropped.

- For "simple" problems with few products and little unbalance between supply and demand, these rules also offer solutions of acceptable quality. However, the branch-first-to-zero rules fail as soon as the size and complexity of the problem increase since, in general, more products or a higher degree of unbalance in the system signal that more

depots have to be open.

- The Y rule is a perfect illustration of the two preceding observations, even if it allows to branch first to either zero or one. It is the most time efficient criterion and its use offers solutions of relatively good quality. This follows from the fact that, most of the time, the Y rule behaves similarly to the rules that close the selected free depot, since all free variables, with the exception of at most one, are less than 0.5 (since they are all non-negative and their sum is less or equal to 1).

- The branching rules that open the selected free depot (i.e. set $y_j' = 1$; rules SMY, $LM\Delta$, $LM\Omega$, SQ, SMQ, SZ) generally obtain the best solutions, as measured by the relative gap. Moreover, these methods are more "robust", in the sense that by using them we consistently managed to obtain an optimal feasible solution in the computational effort allowed.

- Among these branching rules, two quotient criteria perform best: the *Smallest Modified Quotient (SMQ)* and the *Smallest Z (SZ)* rules. However, based on the results of this experimentation, it is not possible to correctly discriminate among these two rules.

In conclusion, it seems that a rule that first branches to zero, such as the Y rule, should be used for simple problems with very few products. In all other cases, either one of the SMQ or the SZ branching criteria should be used.

We also examined the question of how to combine the efficiency of the branch-first-to-zero criteria with the quality of the quotient rules.

The most promising approach consists of building *hybrid* rules that branch to zero (and thus simplify the network and accelerate the procedure) for some of the free variables, at some "appropriate" instances (this should mostly happen in the first stages of the resolution), while using a good quotient rule most of the time. The switch from the zero branching rule to the quotient rule is controlled by a *threshold* on the value of a certain measure associated with each free depot. Of the alternatives that were explored, two offered the most interesting perspectives.

In the first approach, one examines the value of the y variable associated with the selected free depot, and one decides to close it (branch first to zero) if this value is "too"

Problem	*SSQ–ε*	*SMQ–ε*	*SZ–ε*	*SSQ–M*	*SMQ–M*	*SZ–M*
Z1P1	13.80	13.80	13.80	14.20	14.20	14.20
Z1P2	29.50	29.50	29.70	28.50	28.50	28.50
Z1P3	13.80	13.80	14.00	13.60	13.60	14.00
Z1P4	7.60	7.60	7.60	7.60	7.60	7.60
Z1P5	7.00	7.00	7.00	7.00	7.00	7.00
Z2P1	46.90	46.90	46.90	46.90	46.90	46.80
Z2P2	23.70	23.70	23.70	23.70	24.70	24.00
Z2P3	10.00	10.00	10.00	9.20	9.20	9.20
Z2P4	7.5	7.5	7.5	7.5	7.5	7.5
Z2P5	8.3	8.3	8.3	8.3	8.3	8.3
Z4P1	40.80	40.80	40.80	41.60	41.60	42.20
Z4P2	24.2	24.2	24.0	23.2	23.0	24.2
Z4P3	9.60	9.60	9.60	9.60	9.60	9.60
Z4P4	11.70	11.70	11.70	11.30	11.30	11.30
Z4P5	6.10	6.10	6.10	6.10	6.10	6.30

Table 4. Performance of Composite Branching Criteria - Relative Gap (%)

small:

- *ε–threshold* : If $y_{j*} = \min_{j \in K_{01}} y_j' < \varepsilon$, then select the $j*$ free depot and explore first the branch where this depot is closed.

A suitable value for the threshold is $\varepsilon = \frac{1}{2m}$, where m is the cardinality of D, the set of candidate depots. Note that this value equals half the average of the values of the y variables in the initial relaxation of the problem, at the root of the branch-and-bound tree.

The second threshold examines the ratio between the fixed cost associated with a free depot and the value of the total flow between this depot and all clients, and closes the depot if this ratio is "too" high:

- *M-threshold*: If $F_{j*} = \max_{j \in K_{01}} \{f_j / y_j'\} > M$, then select the $j*$ free depot and explore first the branch where this depot is closed.

Several values are possible for M. After some testing, we retained a formula which combines the highest and the average fixed costs of the candidate depots:

$$M = m \max_{j \in D} f_j + \sum_{j \in D} f_j .$$

Problem	*SSQ–€*	*SMQ–€*	*SZ–€*	*SSQ–M*	*SMQ–M*	*SZ–M*
Z1P1	652	650	768	403	251	786
Z1P2	496	496	97	136	213	259
Z1P3	74	98	350	366	424	297
Z1P4	629	785	969	34	244	130
Z1P5	597	713	865	437	955	863
Z2P1	288	298	428	342	477	882
Z2P2	790	839	920	879	515	941
Z2P3	514	531	659	561	621	747
Z2P4	607	559	889	549	563	849
Z2P5	39	235	207	39	235	207
Z4P1	318	326	829	397	459	977
Z4P2	327	181	775	505	737	955
Z4P3	34	310	430	29	195	213
Z4P4	28	50	120	30	70	50
Z4P5	577	795	789	37	973	842

Table 5. Performance of Composite Branching Criteria - "Optimal" Iteration

The specification of the M threshold suggested the definition of a new branching rule. It is based on the observation that the two "best" rules, in terms of quality of solution, are the quotient rules that consider the fixed depot costs and that branch first to 1. The new rule thus compares the volume of the most "profitable" traffic, the one between clients and depots, to the "unprofitable" cost of operating the depots:

Smallest Simplified Quotient Rule (SSQ). Select the free depot which has the minimum F_j, where

$$F_j = f_j \,/\, y_j' \qquad \forall\, j \in K_{01},$$

and explore first the branch where the depot is open.

The two threshold rules were then combined with the the three branching criteria, *SSQ*, *SMQ* and *SZ*, to form six new hybrid, or *composite*, branching rules. The results of the experimentation concerning these rules are illustrated in Tables 4, 5 and 6. Each composite branching criteria is identified by the juxtaposition of the symbols identifying the quotient selection rule and the threshold measure.

The performance measures used in the first experiment, the relative gap between the ''optimal'' solution and the initial lower bound, the iteration where this solution has been found, and the total computing times to treat a node and to select the next variable, are also used to compare the composite branching rules. Problems of medium size, with varying degrees of demand/supply unbalance and with one to five different products, were used for testing; typical results for fifteen such problems appear in Tables 4 to 6. Most of the test problems were used in both experiments, and appear in all six tables, which permits meaningful comparisons among all branching criteria.

Problem	*SSQ–€*	*SMQ–€*	*SZ–€*	*SSQ–M*	*SMQ–M*	*SZ–M*
Z1P1	541.34	541.34	551.07	576.00	585.58	586.63
	0.16	4.75	40.83	0.15	4.56	38.50
Z1P2	935.60	919.68	891.10	939.14	939.14	913.13
	0.16	13.04	108.61	0.17	11.82	97.68
Z1P3	1653.04	1649.45	1654.71	1721.71	1720.17	1775.05
	0.16	14.39	167.03	0.16	14.32	160.62
Z1P4	2153.12	2099.68	2068.66	2370.26	2138.25	2110.63
	0.17	29.54	271.03	0.17	27.20	250.90
Z1P5	3016.97	2984.58	2927.76	3376.07	3373.64	3369.55
	0.16	45.04	400.28	0.15	26.58	275.68
Z2P1	445.62	450.36	461.00	454.50	455.80	467.83
	0.16	5.38	50.73	0.17	5.66	52.25
Z2P2	902.61	879.02	929.82	907.98	997.93	981.31
	0.15	10.43	92.02	0.16	11.20	87.45
Z2P3	1185.89	1200.78	1186.61	1382.97	1400.85	1415.53
	0.17	18.21	166.50	0.17	13.25	133.02
Z2P4	2254.46	2260.00	2218.36	2338.55	2300.54	2264.90
	0.16	34.37	317.83	0.17	35.59	324.09
Z2P5	2882.15	2889.62	2833.34	2884.22	2889.51	2837.16
	0.17	35.78	324.12	0.17	35.78	328.04
Z4P1	548.71	548.72	547.84	633.73	633.39	599.41
	0.17	7.09	55.60	0.17	6.35	41.26
Z4P2	1048.36	1102.24	1113.97	1113.55	1111.17	1031.32
	0.16	11.62	83.37	0.16	9.02	95.98
Z4P3	1780.24	1647 92	1681.66	1733.08	1549.44	1550.39
	0.16	19.72	212.77	0.17	23.71	229.82
Z4P4	2332.37	2293.27	2322.21	2343.58	2296.08	2347.18
	0.17	26.90	380.22	0.18	42.37	392.94
Z4P5	2879.71	2827.20	2807.23	3085.68	3014.74	3013.43
	0.17	51.70	443.51	0.16	34.82	354.29

Table 6. Performance of Composite Branching Criteria - Computing Times (CPU sec.)

The performances of the branch-and-bound algorithm using the six composite branching rules appear roughly equivalent. Yet, consistent differences may be observed and patterns emerge:

- All composite selection criteria consistently outperformed the simple ones, in all measures: solution quality, search speed, computing efficiency.
- Regarding specifically the quality of the solutions obtained as measured by the relative gap between the best solution and the initial lower bound, the three composite criteria behave almost identically: in general, all perform increasingly better when the complexity of the problem increases with the number of products and the degree of supply/demand unbalance. The modified and simplified quotient criteria with the M-threshold seem, however, to perform marginally better.

	Z1			Z2			Z3		
	SSQ	*SSQ*-ε	*SSQ*-*M*	*SSQ*	*SSQ*-ε	*SSQ*-*M*	*SSQ*	*SSQ*-ε	*SSQ*-*M*
P1	13	11	11	28	9	9	37	17	17
	25	33	33	41	77	89	77	115	115
P2	46	27	35	31	13	13	26	14	14
	57	49	53	83	129	129	65	91	103
P3	13	9	40	36	17	17	37	24	13
	89	121	121	99	109	109	145	217	253
P4	33	10	10	28	20	20	44	18	11
	53	111	111	89	141	141	107	113	143
P5	12	9	9	18	18	13	17	12	12
	29	33	33	85	85	113	55	79	79
P6	12	21	41	48	43	93	17	39	39
	37	63	95	71	87	159	41	79	79
P7	16	14	18	11	9	9	13	9	9
	35	103	131	29	53	53	39	71	71
P8	10	9	37	9	8	8	14	10	57
	39	61	99	21	27	27	73	133	185
P9	10	30	16	9	35	35	20	69	77
	39	63	51	19	41	41	73	123	137
P10	12	45	29	16	36	36	10	22	37
	35	71	71	21	107	107	25	57	83

Table 7. Optimal Iteration and Total Enumeration

- The six criteria perform predictably with respect to computing times. These increase with the size and complexity of the problems and with the complexity of the formulas that have to be computed for the threshold and the quotient selection rule.

- The composite branching rules using the M-threshold, generally find the ''optimal'' solution at an earlier stage of the resolution process.

- The simplified quotient rule with the M-threshold, marginally seems to offer the best overall combination, with the modified quotient rule with the M-threshold a solid second.

In order to assess more finely the role of the threshold in the performance of the composite criteria, we recorded for thirty small size (50 nodes, 8 candidate depots) problems the iteration at which the optimal solution is found (the ''Optimal Iteration''), and the total number of iterations required to explore the branch-and-bound tree and prove the optimality of the solution (the ''Total Iterations''). The results for the simplified quotient rule are displayed in Table 7.

Two main conclusions emerge. First, the inclusion of a threshold in the branching criteria seems to have as principal result that the optimal solution is found at an earlier stage than otherwise. The price for this speed up is that a larger number of iterations is generally required to prove optimality. The second conclusion is that when the problems display higher degrees of supply/demand unbalance or larger number of products, thresholds play a lesser role and may even be unproductive.

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Z1	36.20	13.80	15.40	9.40	9.90	4.90	5.70	7.00	4.70	2.40
Z2	19.60	15.90	13.20	8.30	10.80	6.00	5.50	4.60	7.20	2.20
Z3	27.50	14.40	15.30	13.00	6.50	6.40	5.40	7.80	6.20	3.60

Table 8. Lower Bound Gap (%)

Thus, our general conclusion is that the smallest simple or the smallest modified quotient rule should be used to select the next free variable and to branch. Composite branching rules, that use these criteria and the M-threshold, should be used when the problem has not too many products or too high a unbalance degree and when one aims at attaining rapidly a very good (optimal) solution, without necessarily proving optimality.

It is clear that this last option will have a better performance when the lower bound is tight. We examined, therefore, the quality of the lower bound obtained by the relaxation described in Section II. Table 8 shows, for the same thirty problems, the differences between the value of optimal solution and that of the lower bound. These results confirm our previous observations: the quality of the bound increases with the complexity of the problem, especially the number of products. In particular, it is extremely satisfactory for problems with high number of products and that require significant interdepot movements to balance the supplies and demands. As most real problems are of this type, the bounding procedure that we propose is entirely satisfactory.

To complete our experiments, we used the proposed branch-and-bound algorithm to solve the location–allocation problem defined in an actual application. The original application considers the planning of the land transportation operations of a major, Europe based, container shipping company. The problem consists of 12 aggregated types of products, close to 300 aggregated customer zones, 130 candidate depots, in a geographical zone covering five western European countries. Once the 9 initial transportation modes have been aggregated into the least cost (eventually multimodal) arcs between each pair of nodes, we obtained a network that contained 5130 arcs. A detailed analysis of this case may be found in Dejax, Crainic and Delorme (1988). In this paper, we concentrate on the computational performance of the branch-and-bound algorithm when used to solve this problem.

	Branching rule	
	SSQ	SMQ
Relative Gap (%)	1.00	1.02
Optimal Iteration	913	809
Computing Times	19668.96	21497.93
(CPU sec.)	1.84	801.96
Num. Open Depots	50	52

Table 9. Performance of Branching Criteria on Actual Application

Table 9 summarizes the performance of the branch-and-bound algorithm, for the same four measures and 1000 iterations on a SUN 3/50 work-station, with the smallest simplified and modified quotient rules, respectively. These figures confirm the other results presented in

this section and demonstrate that the branch-and-bound algorithm is an efficient solution method for this type of problem.

- The two branching criteria are quite equivalent: the solutions determined by using either one of them are of roughly the same quality. The method using the simplified quotient rule finds a slightly better solution, but at a latter iteration.
- The lower bound is very tight.
- Thresholds are not necessary. In fact the use of a threshold impedes the efficiency of the algorithm: by using a combined rule simplified quotient with M-threshold no integer solution was found in 1000 iterations. Note that for this problem, the two thresholds are equivalent, since all fixed costs are of equal value.

Thus, for problems of real size and complexity, the bounding procedure that we propose is very good, and the smallest simple or modified quotient rule should be used to select the next free variable and to branch.

CONCLUSIONS

We have presented a branch-and-bound algorithm for the location–allocation problem with interdepot balancing requirements. We also report the results of several series of tests that we have conducted, both on actual and on randomly generated problems of various sizes and different degrees of complexity. These tests assisted us to determine the most efficient configuration (branching rule) of the branch-and-bound algorithm for the problem defined by a real application, and to identify the behaviour of the algorithm on more general network configurations.

We have also shown that, while branch-and-bound is an efficient solution method for location problems with balancing requirements, classical results on the design of the algorithm do not apply, and that the special structure of the problem requires special rules and has a determining influence on algorithmic performance. This is particularly true concerning the behaviour of the bounding procedure: the weak relaxation of the problem generates a problem which is easy to solve (minimum cost flow formulation) and which gives a very tight bound.

ACKNOWLEDGMENTS

We want to thank three of our students which have greatly assisted us with the coding of the algorithms and the experimentation: Miss Nicole Tétreault, Mr Jacques Hodgson Jr. and, especially, Mr Bernard Gendron which has also contributed significantly to the identification and study of hybrid branching rules. We also thank the European maritime container shipping company which kindly provided the data for the real large scale application. Funding for this project has been available due to the National Engineering and Natural Sciences Council of Canada (personal grants), the FCAR fund of Québec (programme des Actions concertées sur le transport des marchandises) and the Québec–France cooperation programs.

REFERENCES

Aikens C.H. (1985), *Facility Location Models for Distribution Planning*, **European Journal of Operational Research** 22, 263-279.

Crainic T.G., Dejax P.J. et Delorme L. (1989), *Models for Multimode Multicommodity Location Problems with Interdepot Balancing Requirements*, **Annals of Operations Research** 18, 279-302.

Dejax P.J., Crainic T.G. et Delorme L. (1988), *Strategic/Tactic Planning of Container Land Transportation Systems*, Joint Meeting **TIMS XXVIII/EURO IX**, Paris, 6-8 juillet.

Hodgson Jr., J. (1989), *Application d'une méthode de "branch-and-bound" au problème de localisation avec transferts entre les dépôts*, M.Sc. Thesis, Publication #622, Centre de recherche sur les transports, Université de Montréal.

Ibaraki, T (1987), *Enumerative Approaches to Combinatorial Optimization*, **Annals of Operations Research**, Volumes 10 and 11, No.1-4.

Khumawala, B.M. (1972), *An Efficient Branch and Bound Algorithm for the Warehouse Location Problem*, **Management Science** 18B (12), 718-731.

Krarup J. and Pruzan P.M. (1983), *The Simple Plant Location Problem: Survey and Synthesis*, **European Journal of Operational Research** 12, 36-81.

Nemhauser, G.L. and Wolsey, L.A. (1988), *Integer and Combinatorial Optimization*, Wiley-Interscience.

ANNEX – PROBLEM GENERATION PROCEDURE

This Annex is dedicated to the presentation of the procedure we designed to generate "random" test problems. The procedure first selects the network, then assigns supplies and demands to clients and, lastly, determines the depot and transportation costs.

The network selection procedure starts from a 263 cities North American road network, which gives for each link the driving distance and time. Two parameters have then to be specified: N, the number of nodes (depots and clients) in the network, and R, the number of zones making up the territory. The boundaries of the zones have also to be specified, thus determining the N_r nodes contained in each. The N nodes are selected by using a $Uniform[1, N+1]$ distribution law, while a $Uniform[1, N_r+1]$ distribution is used to chose the m_r candidate depots, and thus the $n_r = N_r - m_r$ clients in each zone.

The concept of "zones" was introduced, primarily because it parallels existing conditions in the logistic organization of large transportation firms: the "logistic zones" of a depot or distribution center. It also facilitates the specification of how much demand/supply unbalance is present in the network. Inside a zone, interdepot transportation costs are not significantly different from the costs of the back and forth movements between clients and depots. Therefore, inside a zone, transportation is mostly represented by client-depot movements, while interdepot traffic is low. On the other hand, for the longer, interzonal movements, interdepot transportation offers greater efficiency and is used to move the traffic required to equilibrate supply and demand. Thus, when generating test problems, the importance of interdepot traffic increases with the number of zones, which therefore represents the supply/demand *unbalance degree* of the problem.

Supplies and demands are therefore determined by product and by zone, according to the following scheme:

- $\Delta_r = n_r * 500$
- $S_r = Uniform[0, \Delta_r]$
- $D_r = Uniform[0, \Delta_r]$

- if $|S_r - D_r| / S_r < \varepsilon$ Start again
- Adjust S_r and D_r such that $\sum_r D_r = \sum_r S_r$.

To distribute the demand between the clients of each zone, we assumed the following two hypotheses:

Hypothesis 1. In each zone there are two groups of clients:

Group 1: 20% of the clients offer 80% of supply;

Group 2: 80% of the clients offer 20% of supply;

Hypothesis 2. Hypothesis 1 is also true for demand, but not necessarily for the same clients.

These "80%/20%" hypotheses are well known in actual practice and were also verified in the case of the application we have analyzed. The procedure to generate supply (demand) figures for a given product, for the clients of a given zone r, is then:

- Choose $0.2n_r$ clients for Group1 by using a $Uniform[1, n_r+1]$;
- For each client i in zone r:
  - $S_i = Uniform[0, 0.8S_r / 0.2n_r]$ for Group 1;
  - $S_i = Uniform[0, 0.2S_r / 0.8n_r]$ for Group 2;
- Adjust S_i for all clients i of zone r such that $\sum_i S_i = S_r$.

Finally, costs have to be assigned to the various operations performed on the network: the opening and operation of depots (the so-called "fixed" costs), and the interdepot and client-depot transportation costs (so-called "variable" costs). Fixed costs for each candidate depot are computed by using a $Uniform[\bar{F}/2, 3\bar{F}/2]$ distribution, where $\bar{F}$ represents the mean fixed cost specified as a parameter to the procedure.

Transportation costs are linear with distance. To insure that interdepot transportation is more cost effective, for interzonal movements, than direct movements between clients and depots, the unit interdepot arc cost has been fixed to 0.72 of the unit client to depot (depot to client) arc cost.

By using this procedure we are able to generate a large variety of problems that differ by the number of candidate depots and clients, the amount of transportation (the total number of km of the links of the network) required, the number of products, the total volume of traffic, the importance of the supply/demand unbalance degree, the range of fixed and variable costs. We generated several such problems, which may be freely obtained from the authors.